Healthcare Twitter Analytics

Basic Text Mining

```
file = 'Tweets_Celiac_sent.csv'
data = read.csv(file,colClasses = "character")
text = data$content
rm(data)
```

Load the corpus and do basic transforms

```
library(tm)
in_corpus = VCorpus(VectorSource(text))

tx_corpus = tm_map(in_corpus, stripWhitespace)
tx_corpus = tm_map(tx_corpus, content_transformer(tolower))
tx_corpus = tm_map(tx_corpus, removeWords, stopwords("english"))
#tx_corpus = tm_map(tx_corpus, removePunctuation)
tx_corpus = tm_map(tx_corpus, stemDocument)

inspect(in_corpus[1])   # before transformation
```

```
## <<VCorpus (documents: 1, metadata (corpus/indexed): 0/0)>>
##
## [[1]]
## <<PlainTextDocument (metadata: 7)>>
## RT @GlutenFreely: Brief, simple descript of #Celiac, #glutensensitivity &amp; #glutenallergy
. Basic knowledge goes a long way! http://t.co/8WNWâ¦
```

```
inspect(tx_corpus[1])   # after transformation
```

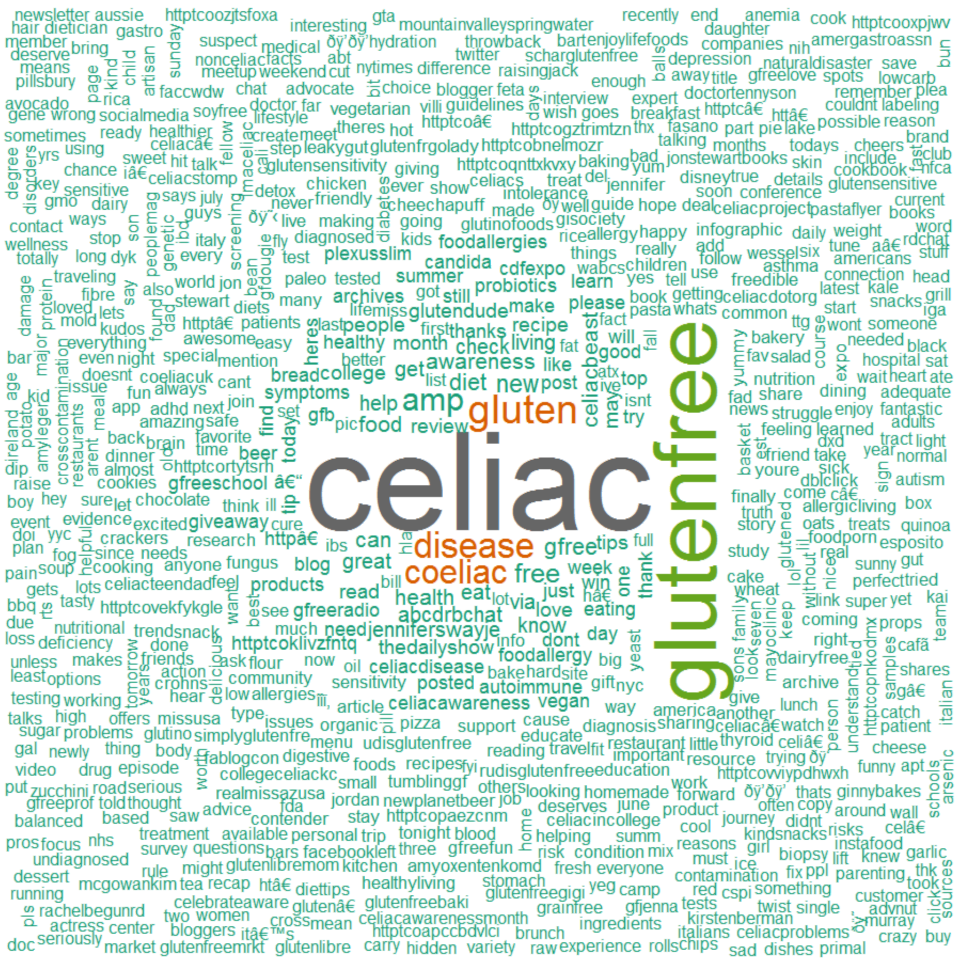
```
## <<VCorpus (documents: 1, metadata (corpus/indexed): 0/0)>>
##
## [[1]]
## <<PlainTextDocument (metadata: 7)>>
## rt @glutenfreely: brief, simpl descript  #celiac, #glutensensit &amp; #glutenallergy. basic
knowledg goe  long way! http://t.co/8wnwâ¦
```

Word Cloud

```
library(wordcloud)

tdm = TermDocumentMatrix(
  in_corpus,
  control = list(
    removePunctuation = TRUE,
    stopwords = c(stopwords("english")),
    removeNumbers = TRUE, tolower = TRUE)
    )

m = as.matrix(tdm)
# get word counts in decreasing order
word_freqs = sort(rowSums(m), decreasing = TRUE)
# create a data frame with words and their frequencies
dm = data.frame(word = names(word_freqs), freq = word_freqs)
wordcloud(dm$word, dm$freq, random.order = FALSE, colors = brewer.pal(8, "Dark2"))
```



Create reduced term matrices

```
dterm_mat <- DocumentTermMatrix(tx_corpus)
dterm_mat <- removeSparseTerms(dterm_mat, 0.95)
inspect(dterm_mat[1:10,])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 9)>>
## Non-/sparse entries: 22/68
## Sparsity           : 76%
## Maximal term length: 10
## Weighting          : term frequency (tf)
##
##     Terms
## Docs #celiac #coeliac #gf #gluten #glutenfre &amp; diseas free gluten
##   1        0        0   0       0          0     1      0    0      0
##   2        1        0   0       0          0     0      0    0      0
##   3        1        0   0       0          0     0      0    2      2
##   4        1        0   0       0          0     0      0    0      0
##   5        1        1   0       0          0     0      0    0      0
##   6        1        1   0       0          1     0      0    1      1
##   7        1        1   0       0          1     0      0    1      1
##   8        0        0   0       0          0     0      0    0      0
##   9        1        1   0       0          0     0      0    0      0
##   10       1        0   0       0          1     0      0    0      0
```

```
tterm_mat <- TermDocumentMatrix(tx_corpus)
tterm_mat <- removeSparseTerms(tterm_mat, 0.95)
inspect(tterm_mat[,1:10])
```

```
## <<TermDocumentMatrix (terms: 9, documents: 10)>>
## Non-/sparse entries: 22/68
## Sparsity           : 76%
## Maximal term length: 10
## Weighting          : term frequency (tf)
##
##             Docs
## Terms        1 2 3 4 5 6 7 8 9 10
##   #celiac    0 1 1 1 1 1 1 0 1  1
##   #coeliac   0 0 0 0 1 1 1 0 1  0
##   #gf        0 0 0 0 0 0 0 0 0  0
##   #gluten    0 0 0 0 0 0 0 0 0  0
##   #glutenfre 0 0 0 0 0 1 1 0 0  1
##   &amp;      1 0 0 0 0 0 0 0 0  0
##   diseas     0 0 0 0 0 0 0 0 0  0
##   free       0 0 2 0 0 1 1 0 0  0
##   gluten     0 0 2 0 0 1 1 0 0  0
```

Find term frequencies

```
findFreqTerms(dterm_mat, 100)  # at least 100 occurences
```

```
## [1] "#celiac"    "#coeliac"   "#gf"        "#gluten"    "#glutenfre"
## [6] "&amp;"      "diseas"     "free"       "gluten"
```

Find correlation to “celiac”

```
findAssocs(dterm_mat, "celiac", 0.7) # 70% correlation
```

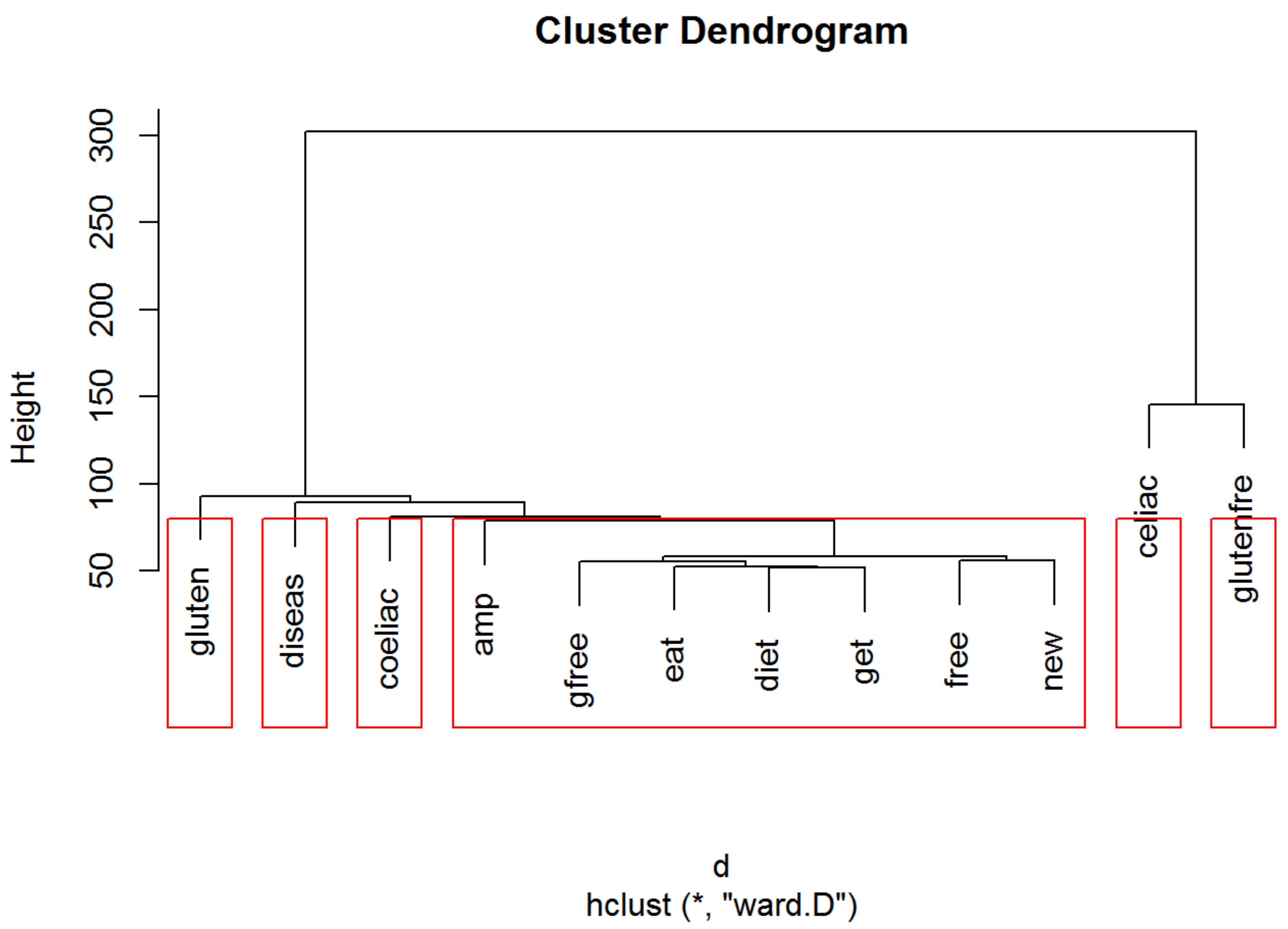
```
## $celiac
## numeric(0)
```

Agglomerative Hierarchical Dendogram

```
# this thing insists on listing its entire contents
#bagofwords = as.data.frame(inspect(tterm_mat))
#saveRDS(bagofwords, file="bagofwords.rds")
bagofwords = readRDS("bagofwords.rds")

bagofwords.scale = scale(bagofwords)
d = dist(bagofwords.scale, method = "euclidean") # distance matrix
fit = hclust(d, method="ward.D")
plot(fit) # display dendogram?

num_clusters = 6
groups = cutree(fit, k=num_clusters)
# draw dendogram with red borders around the clusters
rect.hclust(fit, k=num_clusters, border="red")
```



Cluster Dendrogram
Height
300
250
200
150
100
50
gluten
diseas
coeliac
amp
gfree
eat
diet
get
free
new
celiac
glutenfre
d
hclust (*, "ward.D")